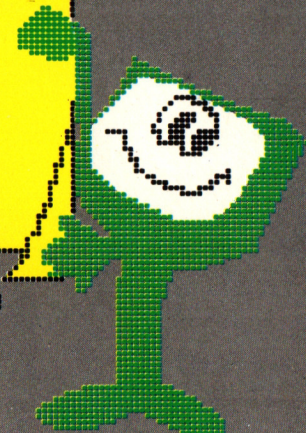
20 VIDEOLEZIONI DI BASIC
PER IMPARARE CON L'MSX

GRUPPO
EDITORIALE
JACKSON

- Trasmissione dati
- Modem e accoppiatori acustici
- Banche dati e reti locali
- Funzioni trigonometriche
- SIN COS TAN ATN
- La gestione degli errori
- Programmazione strutturata
- Videoesercizi
- Videogioco n° 15

VIDEOBASIC MSX
Pubblicazione quattordicinale
edita dal Gruppo Editoriale Jackson
Direttore Responsabile:
Giampietro Zanga
Direttore e Coordinatore Editoriale: Roberto Pancaldi
Autore: Softidea -
Via Indipendenza 88-90 - Como
Redazione software:
Michele Casertelli
Francesco Franceschini
Progetto grafico:
Studio Nuovidea - via Longhi, 16 - Milano
Impaginazione:
Moreno Confalone
Illustrazioni:
Cinzia Ferrari, Silvano Scolari
Fotografie:
Marcello Longhini
Distribuzione: SODIP
Via Zuretti, 12 - Milano
Fotocomposizione: Lineacomp S.r.l.
Via Rosellini, 12 - Milano
Stampa: Grafika '78
Via Trieste, 20 - Pioltello (MI)
Direzione e Redazione:
Via Rosellini, 12 - 20124 Milano
Tel. 02/6880951/5

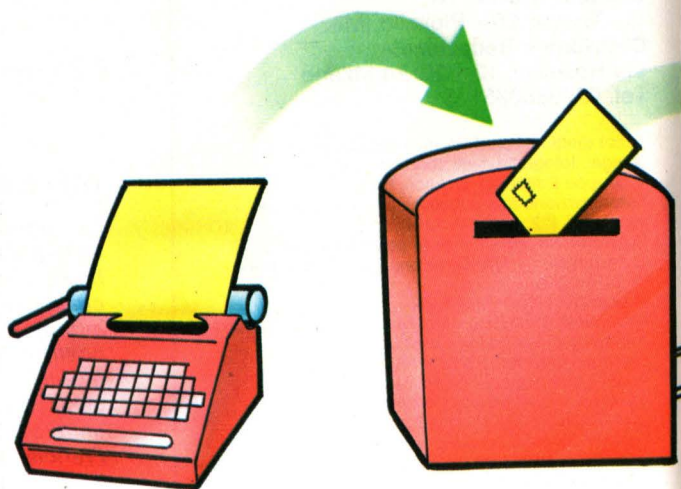
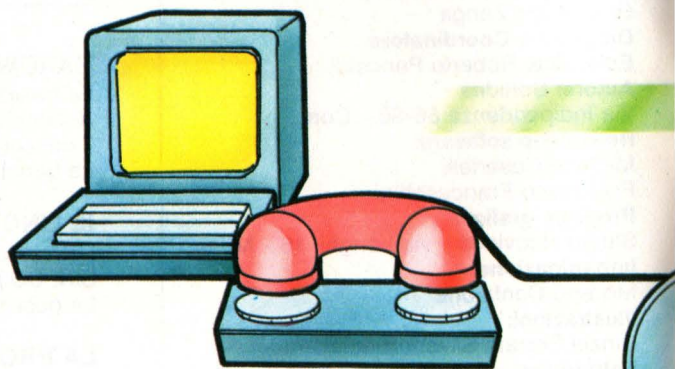
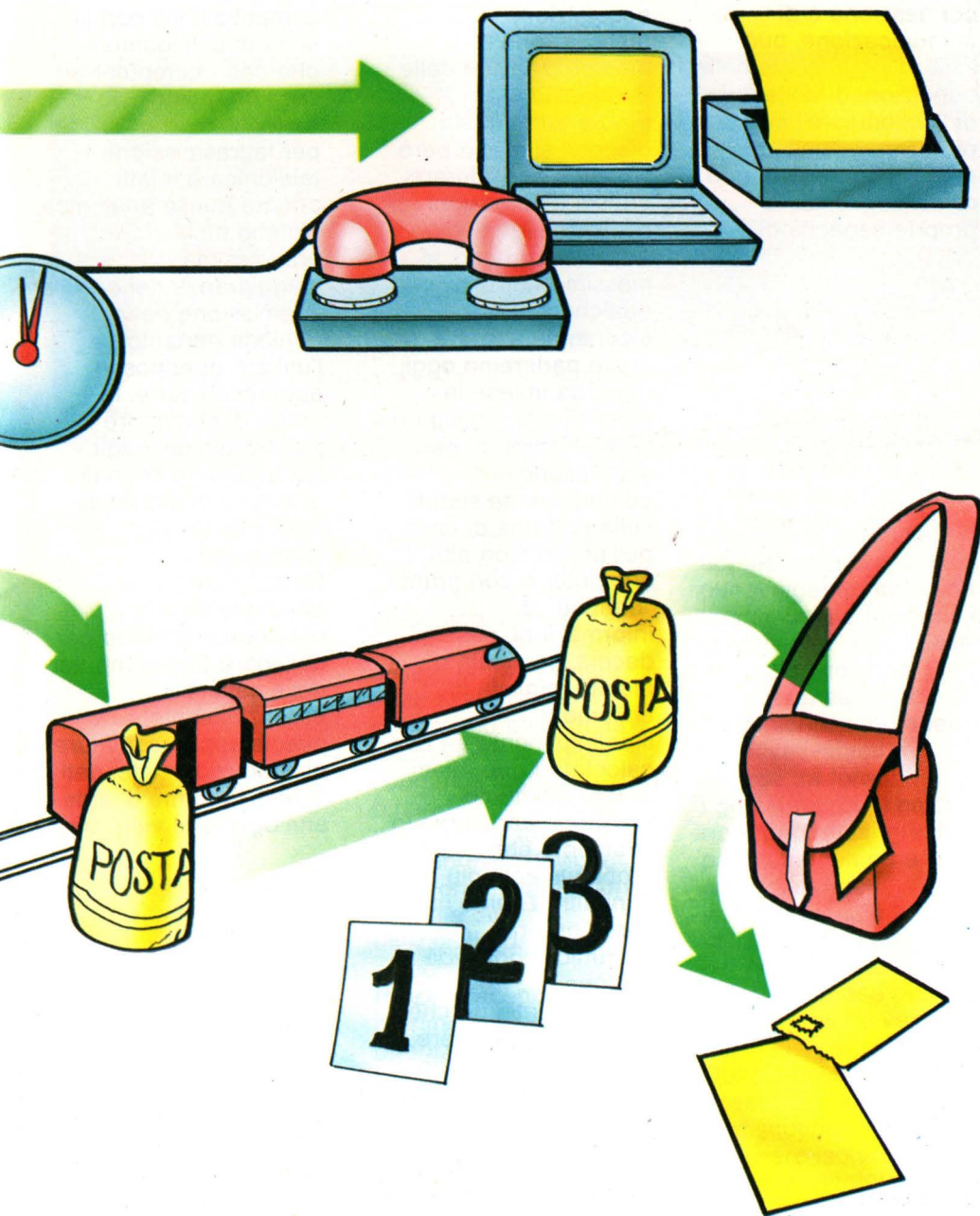
Autorizzazione alla pubblicazione Tribunale di Milano n° 422 del 22-9-1984
Spedizione in abbonamento postale Gruppo II/70 (autorizzazione della Direzione Provinciale delle PPTT di Milano).
Prezzo del fascicolo L. 8.000
Abbonamento comprensivo di 5 raccoglitori L. 165.000
I versamenti vanno indirizzati a: Gruppo Editoriale Jackson S.p.A. - Via Rosellini, 12 20124 Milano, mediante emissione di assegno bancario o cartolina vaglia oppure utilizzando il c.c.p. n° 11666203.
I numeri arretrati possono essere richiesti direttamente all'editore inviando L. 10.000 cdu. mediante assegno bancario o vaglia postale o francobolli.
Non vengono effettuate spedizioni contrassegno.

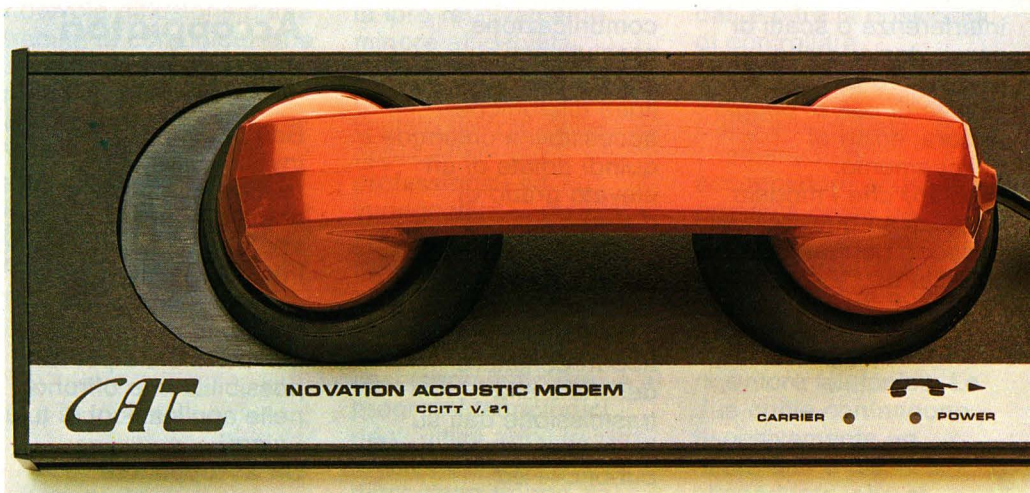
SOMMARIO

Introduzione

La nuova frontiera dei computer è nella trasmissione dati. Un economico home computer, già oggi, può collegarsi, per attingere o scambiare informazioni, con un altro elaboratore posto a mille miglia di distanza. Pensa a poter disporre direttamente sul tuo computer del programma appena utilizzato dal tuo amico di San Francisco. Non male, eh?
Il mezzo lungo il quale avviene il trasferimento delle informazioni è la normale linea telefonica. Dispositivi necessari sono il modem o l'accoppiatore acustico (affidabile e veloce il primo, economico e versatile il secondo). Con poco hardware aggiuntivo, in ogni caso è possibile "espandere" il proprio computer oltre i suoi limiti fisici.

La trasmissione dei dati

Come abbiamo già avuto modo di accennare in altre occasioni, una delle più interessanti e promettenti applicazioni offerte dall'utilizzo dei computer è sicuramente costituita - oltre che dalla pura e semplice capacità di calcolo e di elaborazione - anche dalla possibilità di ricevere e trasmettere informazioni da e verso il mondo esterno. Qualsiasi calcolatore, purché equipaggiato con adeguati ed appropriati dispositivi di

La posta elettronica è un esempio che dimostra quanto risparmio in tempo e risorse sia possibile realizzare per mezzo della trasmissione delle informazioni tramite computer.

HARDWARE

connessione e di comunicazione, può essere infatti posto nelle condizioni di collegarsi e di "colloquiare" con le più diverse "unità periferiche", allargando di conseguenza le proprie capacità di acquisizione, elaborazione e memorizzazione delle informazioni.
Finora tutti i nostri discorsi si erano però limitati a considerare queste connessioni come collegamenti abbastanza brevi, al massimo dell'ordine di qualche metro. La cosa sicuramente nuova, della quale parleremo oggi, riguarda invece la possibilità di collegare il proprio elaboratore, standosene comodamente seduti sulla poltrona di casa o dell'ufficio, con altri computer o con grandi "banche" di informazioni, posti a decine, centinaia o magari migliaia di chilometri.
Il collegamento tra due calcolatori può infatti essere effettuato passando attraverso lo strumento che probabilmente più di ogni altro l'uomo utilizza per le proprie comunicazioni, cioè il telefono.
Naturalmente, perché una cosa del genere sia fattibile è necessario effettuare la trasmissione dei dati seguendo un procedimento in grado di rendere le informazioni trasmesse compatibili sia con la linea di collegamento che con i computer ad essa collegati.
La tecnologia utilizzata per la trasmissione telefonica è infatti ancora quella analogica, almeno nella grandissima maggioranza dei casi; la trasmissione dei dati richiede pertanto l'utilizzo di apposite apparecchiature, in grado di convertire i segnali digitali degli elaboratori in segnali analogici trasmissibili mediante linee telefoniche.
Da ciò deriva che chiunque desideri utilizzare il telefono per collegare due computer deve basarsi essenzialmente su queste apparecchiature di "traduzione" dei dati digitali in forme d'onda analogiche (e viceversa).

Il Modem

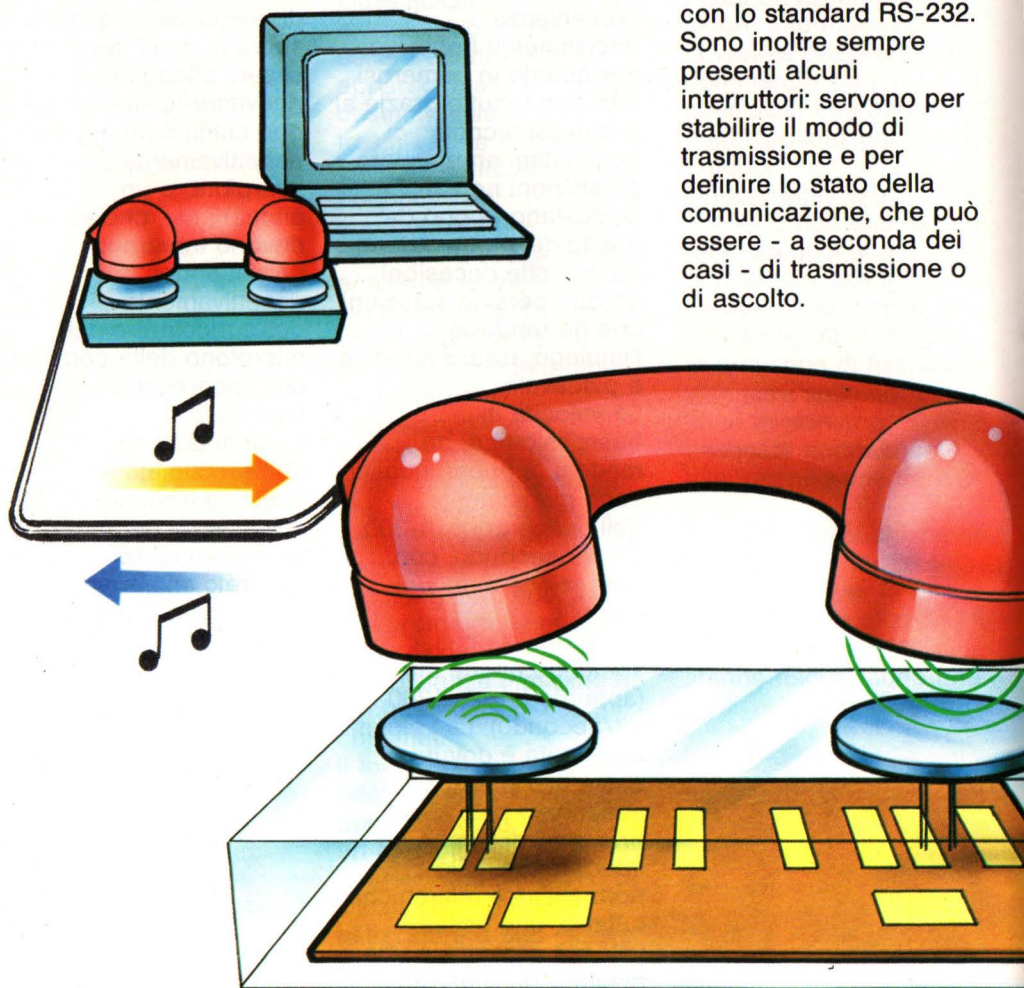
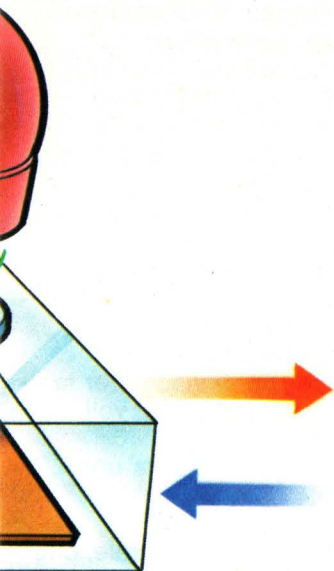
Per poter trasmettere dati digitali attraverso un mezzo di comunicazione analogico, come abbiamo appena visto essere la tradizionale linea telefonica, è necessario che l'unità trasmittente MOduli l'originale segnale binario in un segnale analogico e che l'unità ricevente lo DEModuli, ritrasformandolo in un segnale digitale utilizzabile da un elaboratore.
Proprio facendo riferimento a queste due operazioni fondamentali, le apparecchiature destinate a svolgere entrambe le funzioni sono comunemente chiamate MODEM.
Sotto il termine modem sono però classificate apparecchiature, che, pur svolgendo tutte una fondamentale funzione di "traduzione" da digitale ad analogico (e viceversa) su linee telefoniche, sono spesso significativamente diverse tra loro.
I grossi centri di calcolo ricorrono infatti già da parecchio tempo a particolari linee telefoniche, che vengono utilizzate esclusivamente per le comunicazioni tra computer e che per questa ragione vengono chiamate dedicate.
Queste linee posseggono l'indiscutibile vantaggio di essere isolate in maniera assoluta da tutti quei disturbi,

interferenze o scatti di commutazione che potrebbero provocare perdite di dati o, peggio ancora, errori di collegamento.
È chiaro che i modem installati su queste linee possono permettersi velocità di comunicazione assolutamente impensabili sulle normali linee telefoniche, accessibili a chiunque e quindi dotate di un elevato grado di disturbo.
Una prima grande classificazione dei modem può pertanto essere effettuata tra apparecchiature destinate alla trasmissione dati su linee telefoniche pubbliche ed apparecchiature destinate a svolgere analoghe funzioni su linee dedicate. Queste ultime, definite con il termine di modem diretti, si distinguono abbastanza profondamente dalle prime sia tecnicamente che funzionalmente. In genere queste unità permettono di raggiungere velocità dell'ordine delle migliaia di bit al secondo (bit/s), con distanze massime di poche decine di chilometri.
Il loro uso, riservato a poche e specifiche applicazioni, è tuttavia un capitolo poco interessante dal punto di vista di un normale utilizzatore e può essere quindi tralasciato con tranquillità.

Accoppiatori acustici

Ben più importante è invece il discorso relativo agli accoppiatori acustici. Questi apparecchi costituiscono infatti un argomento di sicuro interesse, sia per la varietà dei prodotti in commercio sia per le possibilità che offrono nelle applicazioni di tutti i giorni.
Un accoppiatore acustico non è altro che un modem, il quale consente a dati e caratteri, trasformati come segnali analogici in una serie di fischi di viaggiare sulle normali linee telefoniche destinate normalmente alla trasmissione della voce.
Questi apparecchi, molto economici e molto interessanti per la loro semplicità di utilizzo e di assoluta trasportabilità, stanno riscontrando una

notevole diffusione come tramite di comunicazione soprattutto da e verso micro e personal computer.
Viceversa, la scarsa velocità consentita dagli accoppiatori acustici (raggiungono al massimo i 1200 bit/s) e la loro relativamente minore affidabilità rispetto ai modem diretti, ne comportano un limitato utilizzo nelle applicazioni più professionali ed impegnative.
I modem acustici tradiscono quindi una provenienza decisamente hobbistica, per quanto in numerosi casi (soprattutto grazie ai progressi tecnologici degli ultimi anni) le loro prestazioni non si discostano troppo da quelle dei modem diretti, ed in molte occasioni offrono persino soluzioni che ne rendono l'impiego assai simpatico e piacevole.
Le velocità di trasmissione di un modem di buona qualità sono mediamente dell'ordine dei 110-300 bit per secondo, con l'eccezione di alcuni modelli semiprofessionali, che si spingono anche oltre (appunto fino ai 1200 bit/secondo). La velocità contenuta è dovuta, come avevamo accennato prima, alla presenza di rumori ed interferenze nelle normali linee telefoniche: tuttavia, una bassa velocità garantisce una migliore ricezione dei dati e offre la possibilità di controllare direttamente la qualità della trasmissione.
Dal punto di vista costruttivo un accoppiatore acustico si presenta come una scatola rettangolare di forma allungata, dotata di due cuffie di gomma sopra le quali deve essere alloggiato il ricevitore telefonico. Le due cuffie contengono rispettivamente un microfono e un altoparlante, che si devono trovare in corrispondenza rispettivamente dell'auricolare e del microfono della cornetta dell'apparecchio telefonico.
L'alimentazione dell'accoppiatore acustico avviene solitamente per mezzo di un alimentatore separato, mentre il collegamento con l'elaboratore va

effettuato con una connessione seriale. La ragione della scelta della tecnica seriale è molto semplice ed intuitiva: dal momento che lungo le linee telefoniche i segnali devono viaggiare sotto forma di "treni d'onda" sequenziali, non esiste infatti alcun'altra soluzione alternativa alla trasmissione seriale. Tutti i modem in commercio sono pertanto seriali e solitamente compatibili con lo standard RS-232. Sono inoltre sempre presenti alcuni interruttori: servono per stabilire il modo di trasmissione e per definire lo stato della comunicazione, che può essere - a seconda dei casi - di trasmissione o di ascolto.

Come avviene il collegamento

In linea teorica - grazie all'accoppiatore acustico - è possibile collegare un qualsiasi tipo di computer con qualunque altro, eliminando in maniera totale eventuali incompatibilità e differenze tra i vari modelli.
Tuttavia, la trasmissione dei dati deve in ogni caso sottostare a un insieme molto rigido e preciso di regole, concernenti soprattutto la codifica e le modalità di invio e di ricezione delle informazioni.
Questo insieme di regole viene comunemente chiamato protocollo di comunicazione.
Dal momento che l'hardware costituisce, almeno in apparenza, un elemento relativamente standard nel campo delle comunicazioni tra elaboratori, il sofware assume quindi un ruolo determinante nell'assestamento definitivo delle incompatibilità che eventualmente fossero ancora presenti tra macchina e macchina.
I numerosi programmi attualmente in commercio, e disponibili per i computer più diffusi, sono pertanto in grado di simulare via sofware molti protocolli di comunicazione, consentendo collegamenti di qualsiasi tipo.
In questo momento il protocollo più utilizzato è il TTY (abbreviazione di TeleTYpewriter, cioè telescrivente). Esso - in un certo senso - ha origini storiche, in quanto ha cominciato ad essere usato proprio con le telescriventi.
Naturalmente, con l'avvento degli elaboratori ha subìto modifiche e migliorie, tendenti soprattutto a una maggiore facilità e velocità di trasmissione, ma nelle sue linee essenziali è rimasto immutato.
Uno dei molti parametri da chiarire prima del collegamento è se il modo di comunicazione viene eseguito in HALF o FULL DUPLEX.
In FULL DUPLEX la

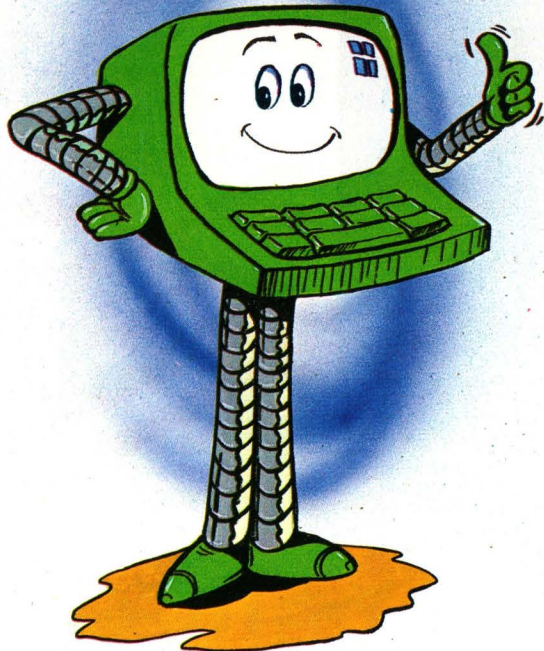
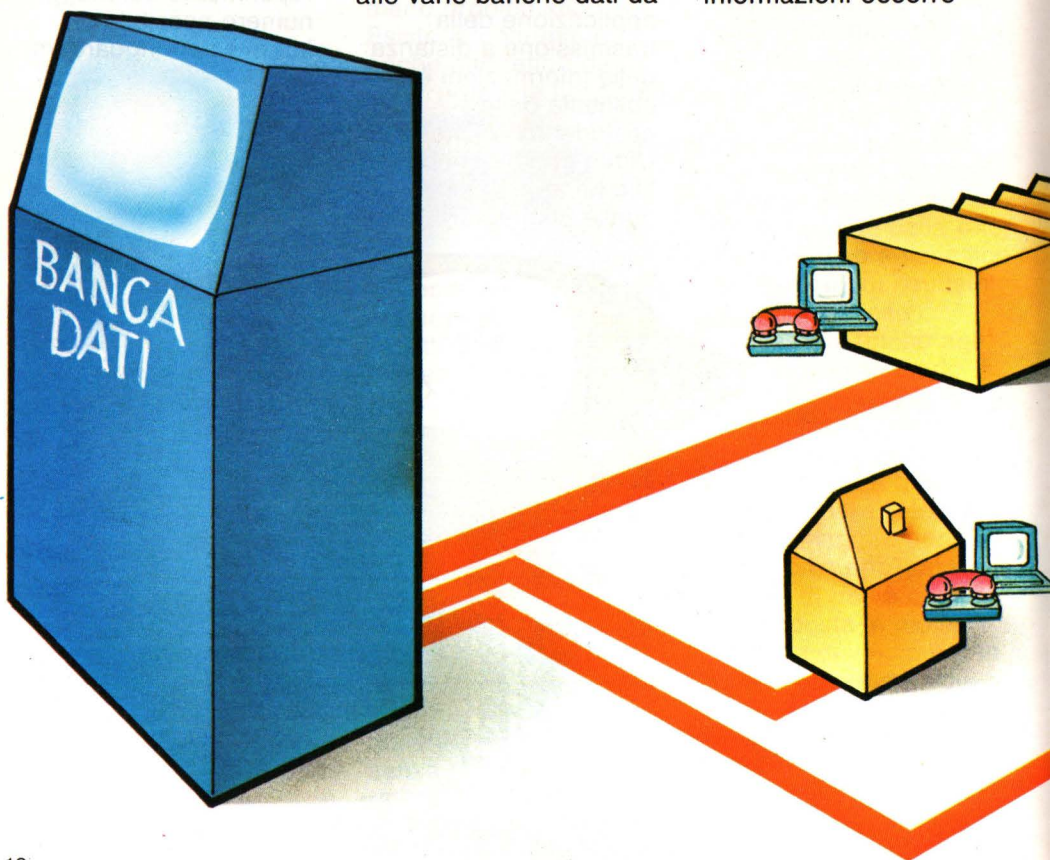
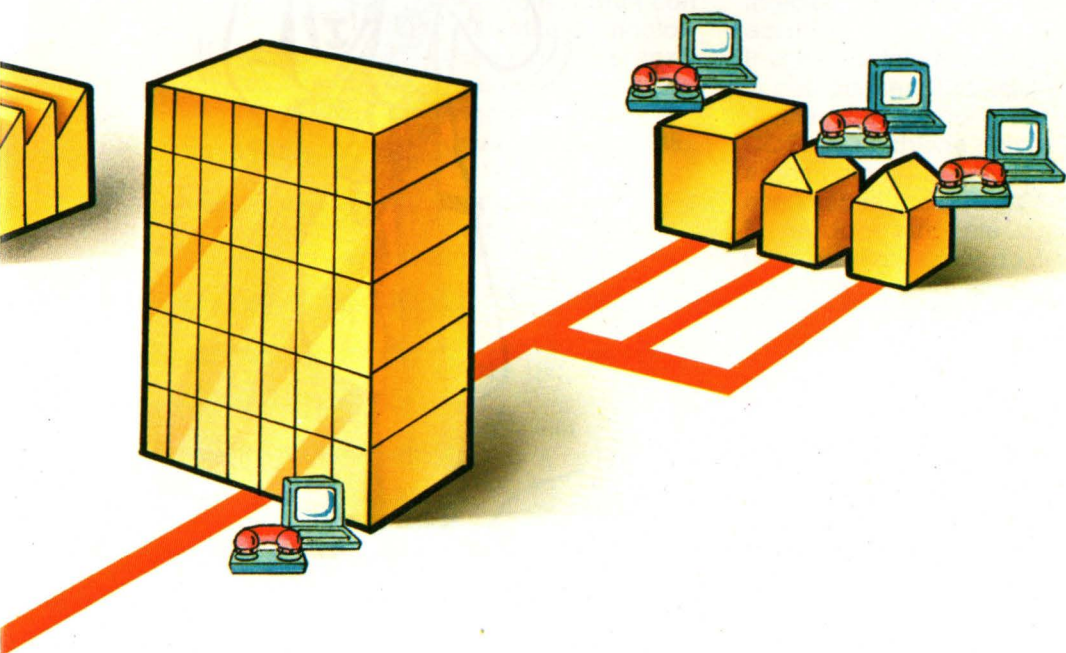
trasmissione tra i modem (e, di conseguenza, tra gli elaboratori) avviene contemporaneamente nelle due direzioni, per cui un carattere trasmesso viene ritrasmesso appena ricevuto e ricompare quindi sul video del primo computer. In questo modo è possibile avere un controllo diretto sulla qualità della trasmissione. In HALF DUPLEX, invece, il carattere non viene trasmesso e quindi non si presta a controlli; naturalmente, è un collegamento molto meno accurato ed impegnativo del FULL DUPLEX, e quindi anche meno problematico.
Esistono comunque anche altri fattori da tener presente nel corso del collegamento: oltre alla velocità e al modo di trasmissione, esistono infatti il controllo di parità (cioè una specie di conteggio dei bit in partenza e in arrivo, che permette di verificare l'esattezza dei dati) e il numero di bit di start/stop che separano ogni singolo carattere.
La cosa migliore è quella di leggere con estrema attenzione il manuale di funzionamento del programma per l'utilizzo del modem: alcune volte accade infatti di voler effettuare collegamenti che non sono materialmente possibili, senza rendersi ragione del malfunzionamento della trasmissione.

I collegamenti in reti

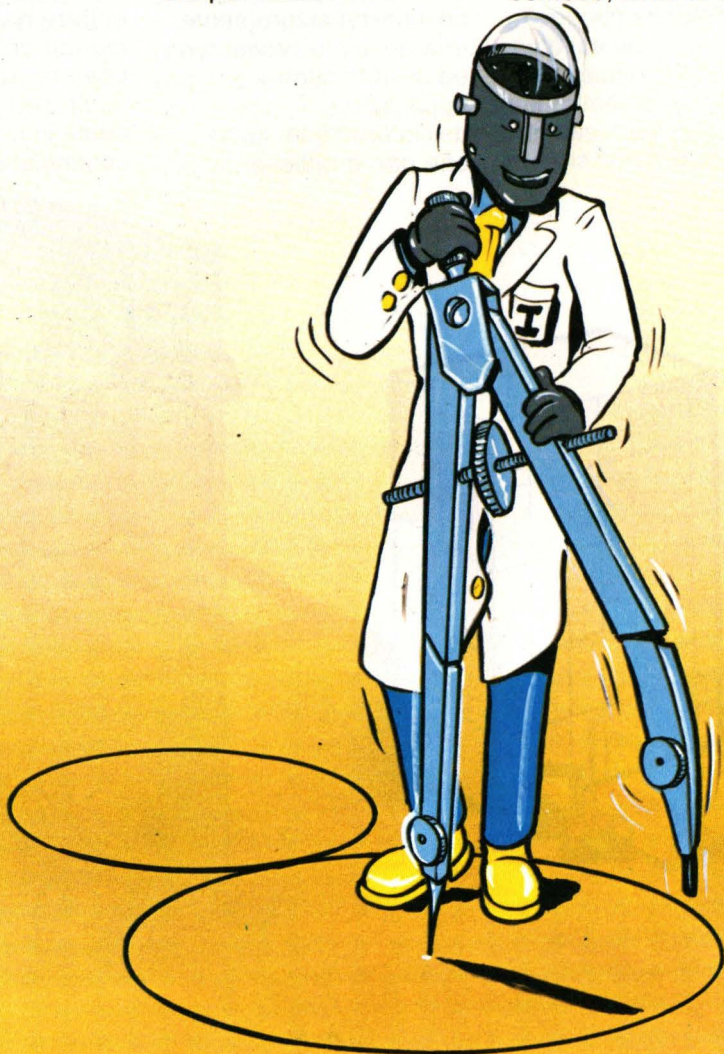
Stabilito una volta per tutte che due elaboratori possono dialogare tra loro attraverso una comune linea telefonica, nulla vieta di considerare l'eventualità di collegare permanentemente, gli uni con gli altri, più elaboratori, così da costituire quella che nel gergo dell'informatica si chiama una rete di dati.
I vantaggi offerti dalle reti di dati sono molteplici: è possibile, per esempio, utilizzare ciascun elaboratore (facente parte della rete) per un unico e specifico compito, il cui risultato può essere reso accessibile anche agli altri. In questo modo ogni computer ha la possibilità di inserirsi e di richiedere le informazioni a qualsiasi estremo della rete, con ovvi vantaggi e benefici.
Una tipica applicazione di rete di dati è quella predisposta in molte agenzie di viaggi per la prenotazione dei voli aerei.
Esiste infatti tutta una rete di terminali collegata a un elaboratore centrale, che

informa sulle disponibilità di posti liberi. Ciascun terminale è in grado di inserire la propria prenotazione nel computer centrale, evitando che in un volo vengano prenotati più posti di quelli effettivamente disponibili. Anche nelle aziende la connessione a rete può risultare estremamente utile: per esempio, i documenti, anziché dover essere trasferiti manualmente da un ufficio all'altro, possono giungere alle persone interessate attraverso gli schermi dei computer collegati, con conseguente risparmio di tempo e di carta.

Le banche dati

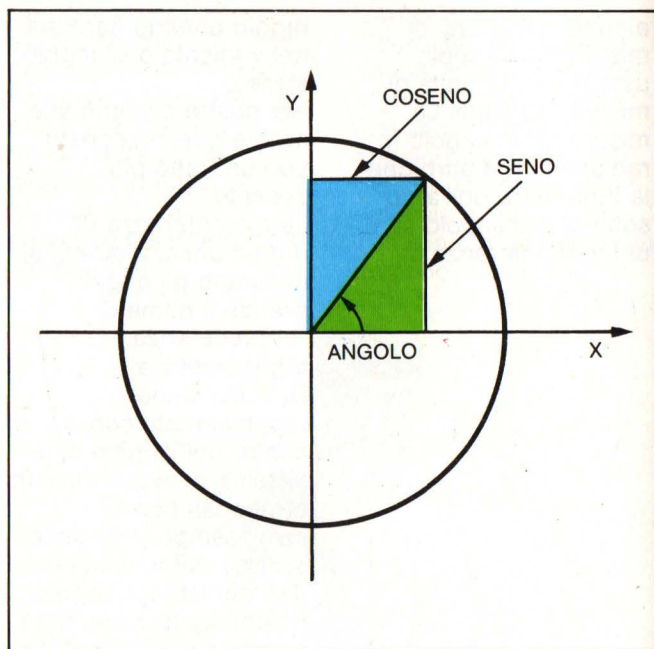
Un'altra utile applicazione della trasmissione a distanza delle informazioni è costituita dalle cosiddette banche-dati. L'idea che ha condotto alle banche dati è stata quella di integrare le grandi capacità di calcolo, ricerca e classificazione di informazioni degli elaboratori con le moderne tecniche di comunicazione tra computer.
Spesso, infatti, nella vita di un'azienda, così come nella vita di uno studioso o di un laureando alle prese con la propria tesi, è necessario giungere in breve tempo al reperimento del maggior numero possibile di informazioni riguardanti

un ben determinato argomento.
Il computer, in una situazione di questo tipo, si pone come lo strumento ideale per risolvere in breve tempo la ricerca, a patto, naturalmente, che sia stata memorizzata da qualche parte una banca di dati (cioè un archivio di informazioni) in grado di rispondere alle esigenze specifiche.
Le prime banche dati nacquero negli Stati Uniti circa 15 anni fa, e da allora molti progressi sono stati compiuti. Allo stato attuale esiste ormai tutto un sistema di reti di computer - estremamente complesso e sofisticato - che permette l'accesso alle varie banche dati da una qualsiasi parte del mondo, attraverso una semplice linea telefonica.
Gli argomenti disponibili sono tra i più vari: medicina, ingegneria, economia, astronomia, biologia...
La connessione con una banca dati è un'operazione molto semplice ed interessante: naturalmente, per l'accesso alle informazioni occorre

pagare alla società fornitrice del servizio un canone di utilizzo, che comunque si giustifica ampiamente con la qualità e la quantità di prestazioni offerte.
Una volta in comunicazione telefonica l'accesso alle banche dati avviene mediante un colloquio fatto di domande e risposte, al termine del quale vengono fornite (sempre che esistano) le informazioni richieste.
In definitiva, la strada della trasmissione dei dati tra computer costituisce sicuramente una delle più promettenti ed avveniristiche applicazioni dell'informatica, tanto che non è difficile prevederne - in un futuro forse nemmeno troppo lontano - un'importante influenza anche nella vita di tutti i giorni.
Per il momento, con una spesa tutto sommato abbastanza accettabile, è possibile per chiunque entrare nell'affascinante mondo delle comunicazioni via computer: l'importante, come si usa dire, è cominciare!

Le funzioni trigonometriche

Se conosci già la trigonometria, probabilmente troverai l'argomento banale, mentre se non ami la matematica potrà sembrarti noioso. In ogni caso abbi pazienza e segui quanto sto per dirti...

L'unità di misura più diffusa per misurare gli angoli, è probabilmente il grado sessagesimale; tutti sanno che un angolo retto misura 90°, mentre un angolo giro equivale a 360°. Tuttavia c'è un modo più comodo, almeno in

alcune occasioni, di misurare gli angoli, usando come unità di misura i radianti. La misura di un angolo in radianti non è altro che la lunghezza dell'arco sotteso dall'angolo su di una circonferenza di raggio unitario centrata nel verticale dell'angolo stesso.
Nel nostro disegno tale arco è quello segnato con un tratto più pesante.
La circonferenza di raggio unitario su cui si misurano gli angoli prende il nome di "circonferenza trigonometrica".
Di solito viene rappresentata con il centro dell'origine di un sistema di assi cartesiani ortogonali con il semiasse positivo delle ascisse coincidente con uno dei lati dell'angolo.
Poiché la circonferenza goniometrica misura 2π, un angolo piatto misurerà π ed uno retto $\pi/2$.
La formula generale per ottenere la misura α in radianti di un angolo a partire da quella A in gradi è:

$$\alpha = A^* \pi / 180.$$

Partendo dalla misura di un angolo in radianti, è possibile risalire alla misura di alcuni segmenti ad esso associati mediante speciali funzioni, dette trigonometriche, che hanno numerose applicazioni nella pratica.
Le funzioni trigonometriche più importanti sono: SENO, COSENO e TANGENTE, e sono tutte calcolabili con il BASIC MSX.
La funzione seno rappresenta la lunghezza del segmento indicato in colore verde.
Il valore di tale funzione si può ottenere in basic mediante la funzione SIN, che richiede come argomento un angolo espresso in radianti.
La funzione coseno rappresenta la lunghezza del segmento indicato in colore azzurro.
Il valore di tale funzione si può ottenere in basic mediante la funzione COS, che richiede come argomento un angolo espresso in radianti.
La funzione tangente rappresenta il rapporto tra il seno e il coseno.
Il valore di tale funzione si può ottenere in basic mediante la funzione TAN, che richiede come argomento un angolo espresso in radianti.
Esistono anche funzioni trigonometriche dette

"inverse", che forniscono la misura dell'angolo a partire da quella del segmento; esse prendono il nome di ARCSENO, ARCCOSENO e ARCTANGENTE.
IL BASIC MSX consente di calcolare solo l'arctangente, mediante la funzione ATN.
Comunque anche le altre funzioni inverse si possono calcolare agevolmente mediante semplici espressioni, che analizzeremo in una delle prossime puntate.
Il valore π, utile in molte occasioni, può essere calcolato mediante l'espressione:

PI = 4*ATN(1)

Il BASIC MSX calcola le funzioni trigonometriche con una precisione veramente ammirevole, superiore a quella di molte calcolatrici scientifiche e di altri linguaggi di programmazione.
Per contro, il calcolo di una funzione trigonometrica risulta relativamente lento, impiegando un tempo attorno a mezzo decimo di secondo.
Questo tempo è irrilevante per semplici calcoli, ma può diventare proibitivo in applicazioni impegnative in cui le funzioni trigonometriche vengono utilizzate migliaia o milioni di volte, o dove la velocità è un fattore rilevante.

NB. Per semplificare useremo il nome PI per pi greco. Ricordati di sostituirlo con la costante numerica 3.1415926535898.

SIN

La funzione SIN consente di valutare il valore del seno corrispondente all'argomento specificato. Così:

```
PRINT SIN (3)
```

stamperà sullo schermo il valore del seno dell'angolo pari a 3 radianti (circa 171 gradi).

Esempi

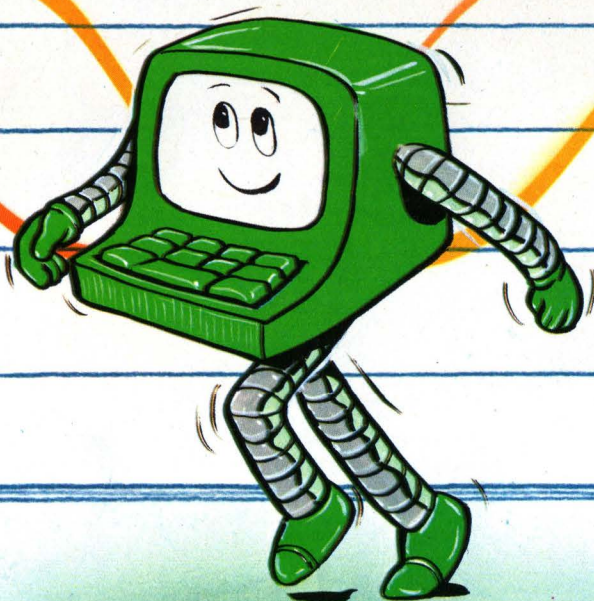
```
PRINT SIN (PI)
PRINT SIN (PI*3)
```

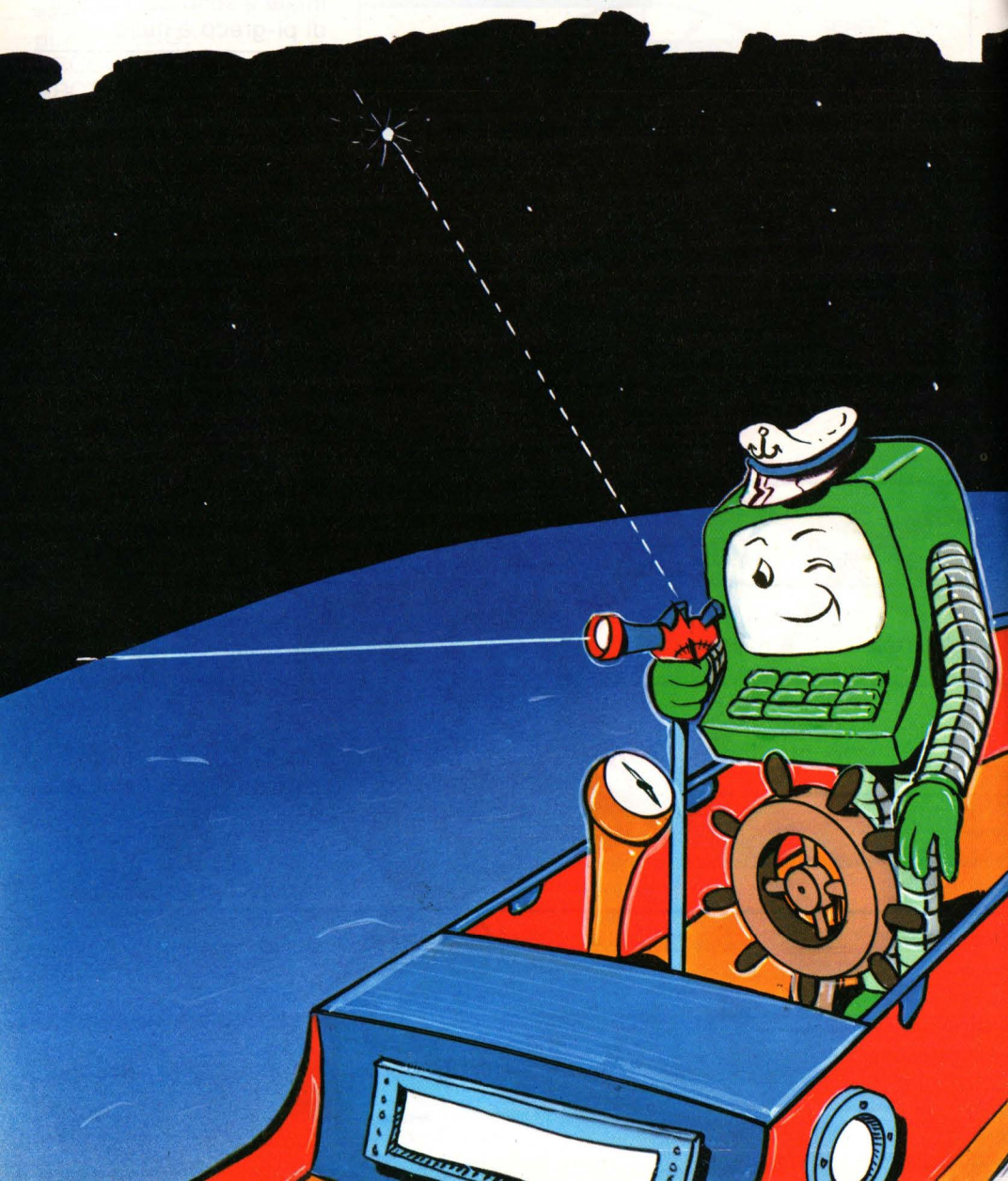
A queste istruzioni il tuo computer risponderà visualizzando 0.
Infatti il seno dei multipli di pi-greco è nullo.

```
PRINT SIN (PI/2)
```

La risposta sarà 1.
L'argomento è infatti il corrispondente in radianti di 90, angolo per cui il seno è massimo

```
LET A = SIN (ANGOLO)
```

Assegna alla variabile A il valore del seno di ANGOLO.

Sintassi della funzione

SIN (valore numerico)

COS

La funzione COS permette invece di valutare il valore del coseno corrispondente all'argomento specificato. Quindi:

```
PRINT COS (3)
```

stamperà sullo schermo il valore del coseno dell'angolo pari a 3 radianti. Naturalmente, questo numero non coinciderà con quello determinato in precedenza da SIN.

Esempi

```
PRINT COS (PI)
```

Mentre il seno a pi-greco radianti è nullo, il coseno risulta massimo. Sullo schermo apparirà quindi il numero 1.

```
PRINT COS (PI/2)
```

Stavolta è il valore del coseno che si azzera!

```
LET SOMMA SIN(C)↑2+COS(C)↑2
```

Prova a battere questa istruzione e a eseguirla con diversi valori di C. Vedrai che il risultato sarà sempre lo stesso, cioè 1.
Questo che abbiamo appena scritto è un teorema fondamentale della trigonometria.

Sintassi della funzione

COS (valore numerico)

TAN

Un'altra funzione trigonometrica è la tangente: essa è definita come il rapporto tra il seno e il coseno di uno stesso angolo. Perciò:

```
TAN(3)= SIN(3)/COS(3)
```

La tangente è una funzione abbastanza delicata: basta infatti poco per vedere che se l'argomento è un multiplo dispari di 90 gradi (od un numero ad esso molto vicino), il computer deve eseguire una divisione per zero (per quegli angoli il coseno diventa nullo). Bisogna quindi fare attenzione a non incorrere in un errore causato da un valore non ammesso.

Esempi

```
PRINT TAN (PI*4)
```

Anche questa volta il risultato sarà 0. Infatti il seno è nullo ed il coseno vale 1. Zero diviso uno è zero.

```
PRINT TAN (PI/2)
```

Stavolta non ci siamo: 1 diviso zero è un errore sicuro!

```
PRINT TAN (B) * COS (B)
```

Per come è stata definita la tangente questa istruzione equivale a scrivere PRINT SIN (B). Infatti (SIN(B)/COS(B))*COS (B)=SIN(B).

Sintassi della funzione

TAN (valore numerico)

ATN

La funzione ATN viene utilizzata per calcolare il valore dell'angolo corrispondente ad una certa tangente.
L'istruzione:

```
PRINT ATN(2)
```

restituirà quindi l'angolo - misurato in radianti - la cui tangente ha valore 2.

Sintassi della funzione

ATN(valore numerico)

La gestione degli errore

Quando il BASIC MSX incorre in una istruzione che non può eseguire, arresta l'esecuzione e segnala l'errore del programmatore con un opportuno messaggio.
In alcuni casi, però, quando si sviluppa o si collauda un programma, si desidera che l'esecuzione continui anche dopo l'incontro di un errore, segnalando l'incorrenza di quest'ultimo in qualche altro modo.

Il BASIC MSX viene incontro alle esigenze del programmatore fornendo una serie di istruzioni atte a gestire gli errori in modo diverso da quello normale.
Queste istruzioni, però, non dovrebbero essere utilizzate in un programma finito, che dovrebbe avere caratteristiche tali a non creare mai condizioni di errore. Ma vediamo insieme quali sono e come funzionano per il trattamento degli errori:

ON ERROR GOTO

Questa istruzione indica al computer che al verificarsi di un errore, invece di arrestare l'esecuzione, deve proseguirla dalla linea indicata dopo GOTO.
In generale la frase ON ERROR GOTO.. viene posta all'inizio del programma, e il sottoprogramma che gestisce l'errore deve essere opportunamente separato dal programma, per non ostacolare la leggibilità.

ON ERROR GOTO 10000

Al verificarsi di un qualsiasi errore, prosegui l'esecuzione dalla linea 10000.

L'istruzione ERROR

Oltre che evitata, l'esecuzione di un errore può essere simulata in qualsiasi momento con l'istruzione error.
Ogni possibile errore è identificato da un numero di codice specifico; i numeri e le corrispondenti frasi si trovano elencati su qualsiasi manuale.
Scrivendo ERROR seguito dal numero di codice di un errore, si provoca la segnalazione di tale errore anche se questo non si è verificato.

```
ERROR 1
```

provoca l'arresto del programma e la segnalazione dell'errore "NEXT without FOR", di codice 1, senza che questo sia realmente occorso.

Le funzioni ERR & ERL

La funzione ERR permette di conoscere il codice dell'ultimo errore occorso; ERL il numero della linea in cui questo si è verificato.
Queste funzioni non richiedono parametri, e si comportano esattamente come variabili, senonché non si possono assegnare ad esse dei valori.

```
PRINT ERR;ERL
```

Stampa il codice dell'ultimo errore occorso ed il numero della linea in cui esso si è verificato.

L'istruzione RESUME

Questa istruzione pone fine al sottoprogramma di gestione degli errori, e torna al programma in esecuzione secondo 3 modalità:
RESUME NEXT - riprende l'esecuzione dall'istruzione successiva a quella che ha causato l'errore.
RESUME 0 - ripete l'istruzione che ha causato l'errore.
RESUME < LINEA> riprende l'esecuzione dalla linea di numero <LINEA>.

ON ERROR GOTO 0

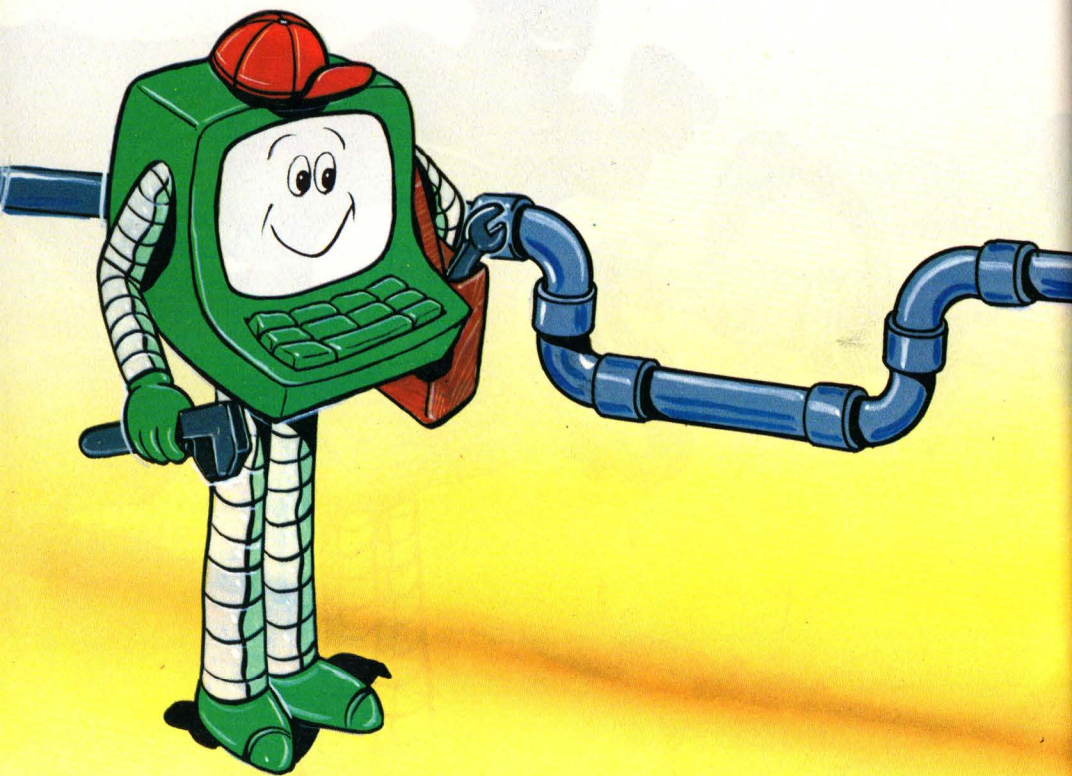
Questa frase serve a interrompere il controllo degli errori da parte di una precedente ON ERROR GOTO...
Dopo la sua esecuzione, il computer riprende a segnalare gli errori nel modo consueto.
L'unico svantaggio è che il sottoprogramma di gestione degli errori non potrà mai cominciare dalla linea 0.

Programmazione strutturata

Uno degli scogli principali nella fase di scrittura di un programma è sicuramente quello provocato dalla fase di studio del problema da risolvere. Più una soluzione è complessa, più ci si trova impegnati per affrontarne tutti i diversi aspetti.
Come già abbiamo detto più volte, l'approccio meno fruttuoso e consigliabile consiste nel sedersi davanti al computer e cominciare a battere linee su linee, via via che sembra di avere raggiunto un certo punto del problema. I risultati, partendo con il piede sbagliato, non possono che essere deludenti.
La via universalmente riconosciuta come migliore è invece quella di procedere con ordine, affrontando la soluzione attraverso una serie di passi elementari descrivibili sotto varie forme, tra le quali la più diffusa è; quella dello schema a blocchi.
Però, nonostante il flowchart rappresenti un utilissimo strumento per descrivere la logica del programma, non esistono delle basi scientifiche sulle quali sia possibile fare affidamento: in altre parole, ogni programmatore applica allo schema a blocchi un proprio insieme di regole, suggerito in genere dalle precedenti esperienze e dalla combinazione di condizioni incontrate fino a quel momento nel singolo problema.
Questo metodo di programmazione, per quanto ampiamente adottato ed utilizzato,

non è per nulla efficiente: le varie parti del problema, anziché essere organizzate in maniera armonica e strutturata, risultano molto spesso intrecciate le une con le altre, rendendo lo schema a blocchi difficile da leggere, capire e modificare.
Nel lavoro professionale (ma anche in quello dilettantistico) tutti questi inconvenienti non possono che tradursi in uno spreco di tempo e di denaro.
È quindi più che giustificabile lo sforzo che ormai da molti anni a questa parte viene rivolto alla ricerca di tecniche di programmazione più pratiche, efficienti ed affidabili.
La trasformazione delle tecniche di programmazione da "arte" a "scienza" risulta tuttavia un problema

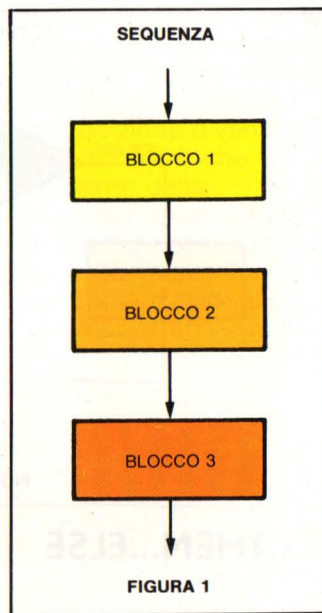
Una cattiva strutturazione porta a programmi complessi e poco chiari. In tali situazioni è molto probabile incorrere in errori difficili da individuare e correggere.

affatto semplice. Attualmente la tecnica riconosciuta come migliore è quella basata sulla cosiddetta "programmazione strutturata", cioè su un insieme di poche e basilari strutture di controllo, che possono venire utilizzate per esprimere qualunque schema logico, non importa quanto complesso, in termini semplici e completi. In questo modo il programma può infatti essere realizzato mettendo insieme blocchi precostituiti, cioè sviluppati indipendentemente gli uni dagli altri. Questa soluzione risolve infatti molti dei problemi appena visti, permettendo una progettazione, leggibilità e modificabilità dei programmi irraggiungibile con altri metodi.
Le tre fondamentali strutture di controllo sulle quali è possibile basare uno schema a blocchi sono:

1) la sequenza;
2) la struttura IF-THEN-ELSE (se...allora...altrimenti);
3) Il ciclo DO...WHILE (esegui...finché).

Vediamole in dettaglio.

Sequenza

Utilizzando la struttura di controllo a sequenza, i blocchi vengono eseguiti uno dopo l'altro. Nella figura 1 i rettangoli rappresentano un singolo blocco da eseguire. Ciascun blocco viene posto nel punto esatto specificato dalla sequenza. È una struttura che possiamo considerare il mattone fondamentale della programmazione.
La conversione in BASIC della sequenza non presenta difficoltà particolari: basta

FIGURA 1

semplicemente tradurre in istruzioni sintatticamente corrette ciascuno dei blocchi facenti parte della sequenza stessa.

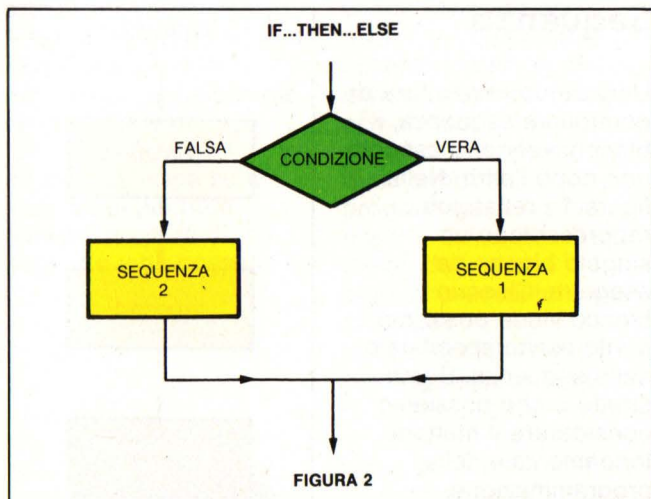



FIGURA 2

IF...THEN...ELSE

La seconda struttura di controllo viene chiamata esecuzione condizionale IF...THEN...ELSE. La parola "IF" della struttura (vedi figura 2) verifica una certa condizione. Se la condizione è vera, allora (THEN) viene eseguita una determinata sequenza, altrimenti (ELSE) ne viene eseguita un'altra. Supponendo che il BASIC MSX non disponga di queste istruzioni vediamo adesso come si potrebbe scrivere in BASIC questa struttura. Se uno dei due rami (l'ELSE) non esiste e la sequenza 1 è breve:

```
50 IF C <> B THEN LET A=B:PRINT "SBAGLIATO"
60
```

Quella che hai appena letto è un'istruzione che ormai conosci molto bene, visto che l'abbiamo usata in moltissime circostanze: la parte della linea che segue il THEN viene eseguita solo se la condizione è vera. Se la parte condizionale è troppo lunga per stare tutta su una linea, si può sempre ricorrere alle subroutine:

```
50 IF A$ = "C"
THEN GOSUB 450 60
```

In questo caso la subroutine alla linea 450 conterrà la sequenza da eseguire quando risulterà vera la condizione.
Se adesso consideriamo la struttura completa (IF...THEN...ELSE), dobbiamo inserire anche la seconda sequenza:

```
50 IF C <> B THEN
    GOTO 90
60 PRINT "GIUSTO"
70 LET C=A
80 GOTO 110
90 LET A=B
100 PRINT "SBAGLIATO"
110
```

Quando la condizione è vera, l'esecuzione passa alle linee 90 e 100 (prima sequenza), da dove poi prosegue indisturbata (linea 110).
Se invece la condizione non è verificata, viene eseguita la seconda sequenza (linee 60 e 70).
La riga 80 evita che al termine della seconda sequenza si entri nella prima.

Do While

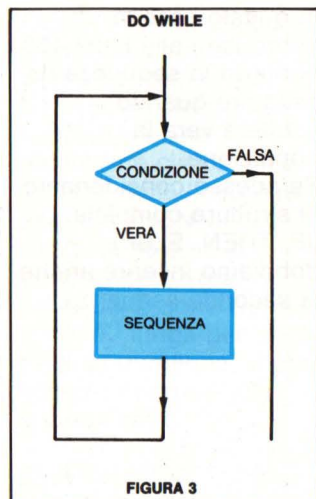
La terza struttura logica, il DO WHILE, viene utilizzata per eseguire e ripetere una certa sequenza fintanto che una determinata condizione risulta verificata. Nella figura 3 puoi vederne il funzionamento.
La condizione viene controllata: se è vera, viene eseguita la sequenza. La stessa condizione viene controllata un'altra volta: se è ancora verificata, la sequenza viene eseguita di nuovo. Questo ciclo continua fino a quando la condizione diventa falsa. È chiaro che all'interno della sequenza deve esserci un blocco che modifichi la condizione, in modo che prima o dopo il ciclo riesca ad avere termine.

Un esempio del ciclo DO-WHILE in BASIC potrebbe essere:

```
80 IF NOT (I<=100) THEN GOTO 120
90 PRINT "SIAMO ARRIVATI A"; I
100 I=I+1
110 GOTO 80
120
```

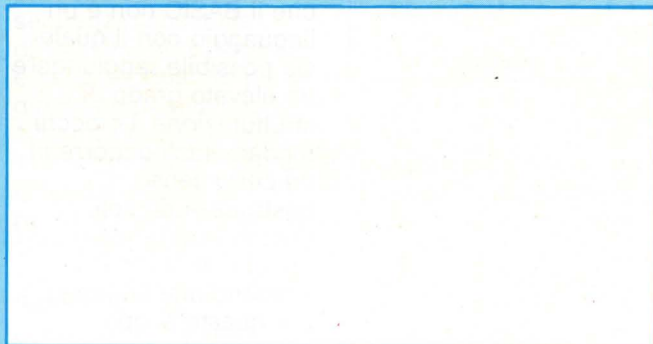
FIGURA 3

Innanzi tutto viene controllata la condizione: se è verificata, si procede con la sequenza (costituita dalle righe 90 e 100); se è falsa, il controllo passa direttamente alla linea 120. Il GOTO alla linea 110 serve per chiudere il ciclo. Da notare la presenza della istruzione 100: è grazie ad essa che il ciclo potrà avere fine, visto che la condizione contiene proprio un controllo sulla variabile I.

Queste tre strutture di controllo possono quindi essere utilizzate per risolvere qualsiasi problema logico di programmazione.
Un concetto importante della programmazione strutturata, del quale non abbiamo ancora parlato, è quello che ogni struttura di controllo dispone di un unico punto di ingresso e di un unico punto di uscita. In questo modo si evitano possibili (e probabili) intrecci delle linee di flusso, rendendo inoltre il programma diviso in moduli indipendenti gli uni dagli altri, e quindi molto più semplici da scrivere e da modificare.

Il modo di operare seguendo la tecnica della programmazione strutturata è tutto sommato abbastanza semplice: si comincia sviluppando uno schema di flusso generale nel quale ciascun blocco (o modulo) rappresenta un'azione (od un insieme di azioni) da eseguire per la soluzione del problema. Si procede quindi per raffinamenti successivi, spezzettando ciascun modulo in una serie di sotto-problemi più piccoli e quindi più semplici da risolvere, giungendo alla fine alla rappresentazione complessiva del problema.
C'è comunque da dire che il BASIC non è un linguaggio con il quale sia possibile raggiungere un elevato grado di strutturazione. I biocchi fondamentali occorre in un certo senso costruirseli da soli, spesso ricorrendo a trucchi ed artifici.
Esistono altri linguaggi, per questo scopo sicuramente più adeguati del BASIC, che consentono - grazie alla loro differente impostazione - di raggiungere lo stesso obiettivo con istruzioni molto più specifiche.
Comunque, la semplicità offerta dal BASIC a giudizio di molti non è ancora stata raggiunta da nessun altro linguaggio.
Con un poco di allenamento scoprirai anche tu quanto la programmazione strutturata possa rendere più chiari e comprensibili i programmi, consentendoti inoltre di abbreviare drasticamente il tempo da dedicare alle fasi di progetto e di correzione degli schemi a blocchi, rispetto alle tecniche meno organizzate e - proprio per questa ragione - meno precise.

VIDEOESERCIZI

Cambiando opportunamente gli esempi proposti da questi esercizi puoi ottenere tabelle e grafici in bassa risoluzione di tutte le funzioni trigonometriche. Attenzione però perché non tutte accettano gli stessi argomenti.

```
10 CLS : PI=3.1415926535898#
20 LOCATE12,0 : PRINT"IL coseno"
30 LOCATE5,2 : PRINT "argomento          coseno"
40 LOCATE5,3 : PRINT "————————       ————————"
50 FORI=—2TO2STEP1/4
60 PRINT"    ";
70 PRINT I; TAB(15);"*pi";TAB(22);INT(100*(COS(I*PI))+.5)/100
80 NEXT I
```

```
10 CLS : PI=3.1415926535898#
20 FORI=—1.5 TO 1.5 STEP.2
30 LOCATE 0,0:PRINTSPC((I+1.5)*9)
40 FOR J=1TO COS(I*PI)*5+12
50 PRINTCHR$(31);
60 NEXT
70 PRINT"*"
80 NEXT
90 LOCATE0,0
```